Compatibility of Safety Properties and Possibilistic Information Flow Security in MAKS

Thomas Bauereiss, Dieter Hutter

Compatibility of Safety Properties and Possibilistic Information Flow Security in MAKS*

Thomas Bauereiss and Dieter Hutter

German Research Center for Artificial Intelligence (DFKI)
Bremen, Germany
firstname.lastname@dfki.de

Abstract. Motivated by typical security requirements of workflow management systems, we consider the integrated verification of both safety properties (e.g. separation of duty) and information flow security predicates of the MAKS framework (e.g. modeling confidentiality requirements). Due to the refinement paradox, enforcement of safety properties might violate possibilistic information flow properties of a system. We present an approach where sufficient conditions for the compatibility of safety properties and information flow security are derived by performing an information flow analysis of a monitor enforcing the safety property and applying existing compositionality results for MAKS security predicates. These conditions then guarantee that the composition of a target system with the monitor satisfies both kinds of properties. We illustrate our approach by deriving sufficient conditions for the security-preserving enforcement of separation of duty and ordered message delivery in an asynchronous communication platform.

1 Introduction

In large, distributed systems that facilitate the collaboration of multiple users there are different types of relevant security requirements. The confidentiality and integrity of data items that are processed in the system needs to be protected, and there are security requirements regarding the users involved in the process, e.g. the requirement that at least two users must agree on a joint decision before the corresponding action can be taken (this requirement is commonly known as separation of duty). Process requirements such as separation of duty can be modeled as safety properties [1]. For confidentiality and integrity requirements, there are various proposals of information flow hyperproperties [5] that go beyond mere access control by taking into account the behavior of the system. The MAKS framework [8], for example, allows to express a range of information flow properties, including several properties proposed in the literature, as a combination of certain basic security predicates.

* This research is supported by the Deutsche Forschungsgemeinschaft (DFG) under grant Hu737/5-1, which is part of the DFG priority programme 1496 "Reliably Secure Software Systems."

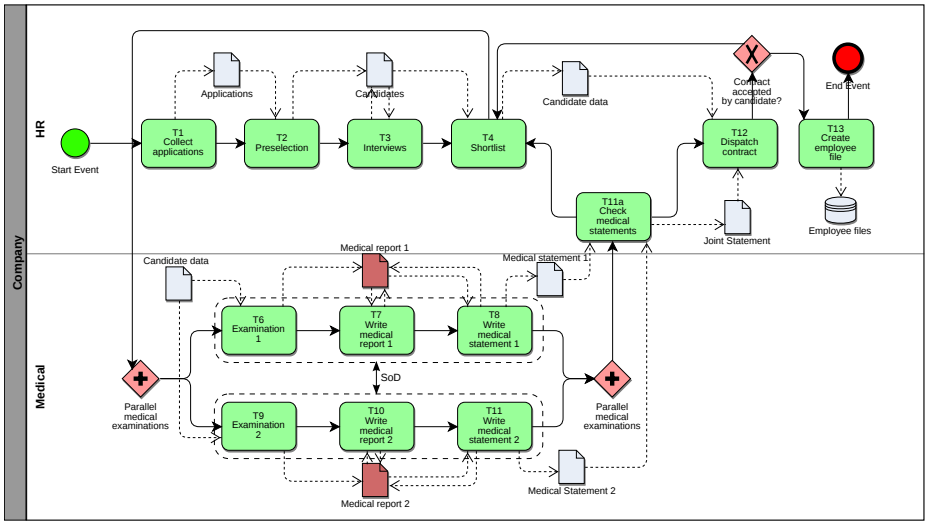

Fig. 1. Example workflow used in [3]

Due to the well-known refinement paradox, the enforcement of a safety property by prohibiting system runs violating it can potentially invalidate possibilistic information flow security: For example, consider a workflow system where a separation of duty constraint between a confidential and a non-confidential activity is enforced. Someone who can observe the non-confidential activity and sees a certain user perform it can deduce that this user has not participated in the confidential activity. This might be an information leak in itself (if anonymity is a concern), and if different users are allowed to perform different actions it might even leak information about the exact sequence of actions that could have been performed in the confidential activity.

In a case study on the verification of information flow security of workflow management systems on an abstract level [3], we considered a hiring process as a running example (Fig. 1). It involves medical examinations of job candidates, and the medical details of these examinations are considered confidential information. We considered two types of separation of duty constraints: We require that the medical examinations must be performed by different persons than the rest of the hiring process due to the need-to-know principle, and we require that there must be two independent medical examinations for each candidate performed by different persons for high assurance of physical fitness of the candidates. The information flows in this example are not entirely trivial, because even though the medical details have to be kept confidential from anyone not involved in the examinations, the final decisions (and only the decisions) must be released to the human resources department so that the workflow can continue. Hence, there is some information flow in the presence of separation of duty constraints, and it is not immediately clear whether there might be subtle interrelations between

confidentiality and separation of duty. This motivated us to formally investigate the compatibility of information flow security and safety properties.

Existing approaches such as [10] on security-preserving refinement can be used to construct a system that satisfies both kinds of properties, but the mechanic modification of the safety property so that it preserves an unwinding relation can lead to unexpected results. We propose to use compositionality [11] for this purpose. A safety property can be enforced using an execution monitor that runs in parallel with the target system and inhibits executions that would violate the safety property. We can analyze such a monitor and verify that it does not leak confidential information under certain conditions, and then compose it with the target system. The composed system satisfies the safety property, and the compositionality theorems of the MAKS framework give us sufficient conditions under which this composition preserves information flow security.

The contribution of this paper is to state this approach formally. It can be applied to arbitrary safety properties, although manual effort seems to be necessary for deriving sufficient conditions for compatibility with information flow security. However, we believe that this manual effort can be very efficient when compatibility results for whole classes of important safety properties are derived. The two example properties that we use for illustration, namely separation of duty and the enforcement of ordered delivery of messages between asynchronously communicating systems, are relevant for many systems, and our results can be instantiated for them simply by replacing the sets of underlying events accordingly. If such a compatibility results exists for a safety property of interest and its side conditions are satisfied, it allows us to prove information flow security for a simplified system that does not need to satisfy the safety property, and then enforce safety by composition with a monitor while preserving security.

The rest of this paper is structured as follows. In Section 2, we recall definitions of state-event systems, information flow security and safety properties from the literature. Section 3 describes our approach of using compositionality for the security-preserving enforcement of safety properties and illustrates it with two examples. Section 4 discusses related work and Section 5 concludes the paper.

2 Preliminaries

2.1 System Model

We briefly recall the definitions of (state-) event systems and security predicates from the MAKS framework for possibilistic information flow [8] that we use in this paper. An event system $ES = (E, I, O, Tr)$ is essentially a (prefix-closed) set of traces $Tr \subseteq E^*$ that are finite sequences of events in the event set E. The disjoint sets $I \subseteq E$ and $O \subseteq E$ designate input and output events, respectively. We denote the empty trace as $\langle\rangle$, the concatenation of traces α and β as $\alpha.\beta$, and the projection of a trace α onto a set E as $\alpha|_E$. In the composition $ES_1 \| ES_2$ of two event systems ES_1 and ES_2, the set of traces is the set of interleaved traces of the two systems, synchronized on events in $E_1 \cap E_2$:

$$Tr(ES_1 \| ES_2) = \{\alpha \in (E_1 \cup E_2)^* \mid \alpha|_{E_1} \in Tr(ES_1) \wedge \alpha|_{E_2} \in Tr(ES_2)\}$$

Input events of one system matching output events of the other system are connected (and vice versa) and thus become internal events of the composed system. Note that we drop the assumption of [11] that all shared events of the two components must be an output event of one component and an input of the other. This allows us to formulate execution monitors for safety properties as event systems with no input and output events of their own, such that the composition of the monitor with a target system retains the input and output events of the original target system. This notion of composition is in line with the generalized parallel composition operator of CSP [15]. All proofs of compositionality of security predicates remain valid, as the concrete sets of input and output events are not used in the proofs at all.[1]

Example 1. In [3], we defined the behavior of workflow systems in terms of the behaviors of communicating subsystems representing individual activities of the workflow. In our example workflow, activities correspond to nodes of the graph in Figure 1. This approach makes the verification simpler and more scalable, as it allows us to use the decomposition methodology of [6] to verify the security of the overall system by verifying security properties of the subsystems. Each activity a is modeled as an event system with a set of events E_a of the form

- $Start_a(u)$, starting the activity a and assigning it to the user $u \in U$,
- $End_a(u)$, marking the end of the activity,
- $Send_a(a', msg)$ and $Recv_a(a', msg)$, representing activity a sending message msg to another activity a' (or a receiving msg from a', respectively),
- $Setval_a(u, i, val)$ and $Outval_a(u, i, val)$, representing a user $u \in U$ writing (or reading, respectively) the value val of data item i during activity a, and
- a set of internal events τ_a.

The behavior of these activities is modeled using internal states S_a and a transition relation $T_a \subseteq S_a \times E_a \times S_a$, inducing the set of possible traces. The overall workflow system $ES_W = (\|_{a \in \mathcal{A}} ES_a) \| ES_P$ emerges from the composition of these event systems ES_a for every activity $a \in \mathcal{A}$, together with a communication platform ES_P. The communication platform asynchronously forwards messages between the activities. Upon composition with the platform, the communication events between the activities become internal events of the composed system. Only the communication events between activities and users remain input and output events. These events form the user interface of the workflow system. □

2.2 Information Flow Security

The MAKS framework defines a collection of basic security predicates (BSPs). Many existing information flow properties from the literature can be expressed as a combination of these BSPs. Each BSP is a predicate on a set of traces with respect to a view $\mathcal{V}$. A view $\mathcal{V} = (V, N, C)$ on an event system $ES = (E, I, O, Tr)$

[1] We verified this using an existing formalization of the MAKS framework for the interactive theorem prover Isabelle. Removing the assumption of matching input and output events has no effect on the validity of the proofs.

$$
\begin{aligned}
BSD_{\mathcal{V}}(Tr) \equiv\ & \forall \alpha, \beta \in E^*. \forall c \in C. \, (\beta.c.\alpha \in Tr \wedge \alpha|_C = \langle\rangle) \\
& \Rightarrow \exists \alpha' \in E^*. \, \big(\alpha'|_V = \alpha|_V \wedge \alpha'|_C = \langle\rangle \wedge \beta.\alpha' \in Tr\big) \\
BSIA^{\rho}_{\mathcal{V}}(Tr) \equiv\ & \forall \alpha, \beta \in E^*. \forall c \in C. \, (\beta.\alpha \in Tr \wedge \alpha|_C = \langle\rangle \wedge \beta.c \in Tr \wedge Adm^{\rho}_{\mathcal{V}}(Tr, \beta, c)) \\
& \Rightarrow \exists \alpha' \in E^*. \, \big(\alpha'|_V = \alpha|_V \wedge \alpha'|_C = \langle\rangle \wedge \beta.c.\alpha' \in Tr\big) \\
FCIA^{\rho,\Gamma}_{\mathcal{V}}(Tr) \equiv\ & \forall \alpha, \beta \in E^*. \forall c \in C \cap \Upsilon. \forall v \in V \cap \nabla. \\
& (\beta.\langle v\rangle.\alpha \in Tr \wedge \alpha|_C = \langle\rangle \wedge \beta.c \in Tr \wedge Adm^{\rho}_{\mathcal{V}}(Tr, \beta, c)) \\
& \Rightarrow \exists \alpha' \in E^*. \exists \delta' \in (N \cap \Delta)^*. \\
& \big(\alpha'|_V = \alpha|_V \wedge \alpha'|_C = \langle\rangle \wedge \beta.c.\delta'.\langle v\rangle.\alpha' \in Tr\big)
\end{aligned}
$$

Fig. 2. The MAKS basic security predicates BSD, $BSIA^{\rho}$, and $FCIA^{\rho,\Gamma}$

is defined as a triple of event sets, where the set V defines the set of events that are visible for an observer, C are the confidential events, and the events in N are assumed to be neither visible nor confidential. A view is *valid* if V, N and C are pairwise disjoint, and it is *valid for* ES if V, N and C form a disjoint partition of E. Notable examples for BSPs, that we will use in this paper, are backwards-strict deletion of confidential events (BSD), backwards-strict insertion of admissible confidential events ($BSIA^{\rho}$), and forward-correctable insertion of admissible confidential events ($FCIA^{\rho,\Gamma}$)[2], defined in [11] as given in Figure 2. Intuitively, BSD requires that the occurrence of confidential events must not be deducible, while $BSIA$ and $FCIA$ require that the *non*-occurrence of confidential events must not be deducible. Technically, they are closure properties of sets of traces. For example, if a trace in Tr contains a confidential event, then BSD requires that a corresponding trace without the confidential event exists in Tr that yields the same observations. This means the two traces must be equal with respect to visible V-events, while N-events might be adapted to correct the deletion of the confidential event.

In [11], compositionality results for these basic security predicates are presented. They give sufficient conditions under which security of a composed system is implied by the security of its subsystems. Let us consider the composition of two event systems ES_1 and ES_2 with event sets E_1 and E_2 and trace sets Tr_1 and Tr_2, respectively. First, the views $\mathcal{V}_i = (V_i, N_i, C_i)$ for the subsystems must form a *proper view separation* of the view $\mathcal{V} = (V, N, C)$ for the composed system, i.e. $V \cap E_i = V_i$, $C \cap E_i \subseteq C_i$ and $N_i \cap N_j = \emptyset$. Second, the components must be well-behaved wrt. the views, i.e. if a shared event is used for corrections in one component, then the other component must accept it at any time without interfering with visible observations. We slightly reformulate the notion of well-behaved composition given in Definition 6.3.6 of [12] as a well-behavedness condition on the individual components to be composed:

[2] The parameters ρ and $\Gamma = (\nabla, \Delta, \Upsilon)$ control at which positions in traces it must be possible to insert confidential events and which corrections are allowed, and admissibility is defined as $Adm^{\rho}_{\mathcal{V}}(Tr, \beta, e) \equiv \exists \gamma \in E^*. \, \big(\gamma.\langle e\rangle \in Tr \wedge \gamma|_{\rho(\mathcal{V})} = \beta|_{\rho(\mathcal{V})}\big)$.

Definition 1. *The component* ES_i *is* well-behaved for $\mathcal{V}_i$ wrt. $\mathcal{V}_j$, *with* $i, j \in \{1, 2\}$ *and* $i \neq j$, *if*

- $N_j \cap E_i \neq \emptyset$ *implies* $total(ES_i, C_i \cap N_j) \wedge BSIA^{\rho_E}_{\mathcal{V}_i}(Tr_i)$, *and*
- $N_j \cap E_i \neq \emptyset \wedge N_i \cap E_j \neq \emptyset$ *implies* $FCIA^{\rho_E, \Gamma_i}_{\mathcal{V}_i}(Tr_i)$,

where $\rho_E((V, N, C)) = V \cup N \cup C$ *and* $\Gamma_i = (E_i \cap E_j, E_i \setminus E_j, C_i \cap N_j)$.

The composition of ES_1 *and* ES_2 *is* well-behaved wrt. $\mathcal{V}_1$ and $\mathcal{V}_2$ *if* ES_1 *is well-behaved for* $\mathcal{V}_1$ *wrt.* $\mathcal{V}_2$ *and* ES_2 *is well-behaved for* $\mathcal{V}_2$ *wrt.* $\mathcal{V}_1$.

Third, specific side conditions for the security predicate in question must be satisfied. For BSD and $BSIA$, this is summarized in the following corollary:

Corollary 1 (of Theorem 6.4.1 in [12]). *Let* $\mathcal{V}_1$ *and* $\mathcal{V}_2$ *be a proper separation of* $\mathcal{V}$ *and let the composition of* ES_1 *and* ES_2 *be well behaved wrt.* $\mathcal{V}_1$ *and* $\mathcal{V}_2$. *Then the following holds:*

- $BSD_{\mathcal{V}_1}(Tr(ES_1)) \wedge BSD_{\mathcal{V}_2}(Tr(ES_2))$ *implies* $BSD_{\mathcal{V}}(Tr(ES_1 \| ES_2))$.
- *If* $BSD_{\mathcal{V}_j}(Tr_(ES_j))$ *and* $\rho_j(\mathcal{V}_j) \subseteq \rho(\mathcal{V}) \cap E_j$ *for all* $j \in \{1, 2\}$, *then* $BSIA^{\rho_1}_{\mathcal{V}_1}(Tr(ES_1)) \wedge BSIA^{\rho_2}_{\mathcal{V}_2}(Tr(ES_2))$ *implies* $BSIA^{\rho}_{\mathcal{V}}(Tr(ES_1 \| ES_2))$.

For details of the compositionality of other basic security predicates, see [12].

Example 2. In our example workflow, we consider the contents of the medical reports as confidential information. Hence, we classify system events representing the input our output of medical reports (i.e. events of the form $\mathrm{Setval}_a(u, i, v)$ and $\mathrm{Outval_a(u, i, v)}$ with $i \in \{\mathrm{MedReport1}, \mathrm{MedReport2}\}$ and a being on the medical activities) as confidential events. The events belonging to activities of the human resources department that do not handle medical information can be considered as potentially visible to an observer. This gives rise to a security view on the overall system, and the security predicates BSD and $BSIA$ formalize the requirement that someone who observes or participates in visible activities cannot deduce information about the occurrence or non-occurrence of confidential events and, hence, the values of confidential data items. See [3] for detailed definitions of the security views and predicates. We used compositionality for the verification of information flow security by applying the methodology of [6] to decompose the overall security property into properties of the subsystems, and verifying those using an unwinding technique [9]. □

2.3 Safety Properties

A safety property can be characterized by a "bad thing" that must not happen [1]. Hence, it can be formalized as the set of traces where this bad thing does not occur. For example, consider a separation of duty constraint between two activities. The bad thing happens when the same user performs both activities.

Example 3. Consider a system that includes several activities to be performed with user interaction, such as our workflow system of Example 1. Let a and a' be two activities between which a separation of duty constraint shall be enforced, for example the medical examinations T6 and T9 in Figure 1. Let E_a and $E_{a'}$, respectively, denote the sets of events belonging to these activities, let E_W denote the set of all events of the workflow system, let U be a set of users, and let E_u denote the events of interaction between user $u \in U$ and the system. Separation of duty between a and a' is represented by the set of traces

$$\{\alpha \in E_W^* \mid \forall u, u' \in U.\, \forall e_1, e_2 \in \alpha.(e_1 \in (E_a \cap E_u) \wedge e_2 \in (E_{a'} \cap E_{u'})) \rightarrow u \neq u'\}$$

It contains only traces where the users participating in a are different from those participating in a'. We denote this safety property as $P_{SoD}^{a,a'}$. □

Such a safety property can be enforced by an execution monitor that is run in parallel with the target system and inhibits executions that would violate the property. Note that the above property is defined solely in terms of events in $(E_a \cup E_{a'}) \cap E_U$, where $E_U = \bigcup_{u \in U} E_u$ denotes the set of all user interaction events. Hence, other events are irrelevant for this property and can be ignored by an execution monitor. This is captured in the following notion of relevant events:

Definition 2. *Let $P \subseteq E^*$ be a safety property, i.e. a set of traces composed of events in E. The set $E_P \subseteq E$ is a* relevant set of events for P *iff for all $\tau \in E^*$ it holds that $\tau|_{E_P} \in P$ implies $\tau \in P$.*

A monitor can then be defined as an event system with a relevant set of events and a set of traces that satisfies the property:

Definition 3. *Let P be a safety property. A* monitor for P *is an event system $ES = (E, I, O, Tr)$ such that E is a relevant set of events for P and $Tr \subseteq P$.*

Composing a target system with the monitor yields a system that satisfies the safety property:

Lemma 1. *Let $ES = (E, I, O, Tr)$ be an event system and $ES_P = (E_P, I_P, O_P, Tr_P)$ be a monitor for a safety property P. Then $Tr(ES \| ES_P) \subseteq P$. Furthermore, if $E_P \subseteq E$, then $Tr(ES \| ES_P) \subseteq Tr(ES) \cap P$.*

This follows directly from the definitions of relevant events, monitor, and set inclusion. For simplicity, we assume below that the set of monitor events is a subset of the events of the target system.[3] In this case, the composed system is a refinement of the original system, in the sense that the set of traces of the composition is a subset of the traces of the original system, and it satisfies the safety property.

[3] Internal monitor events modeling enforcement could be added in a subsequent refinement.

3 Secure Composition with Safety Monitors

Now that we have cast the enforcement of a safety property as a composition of the target system with a monitor, we can leverage compositionality results for information flow predicates to obtain conditions under which the enforcement of the safety property preserves information flow security. Consider the situation that we have a target system that we have already proven secure, but that does not yet satisfy a safety property, and we have a monitor for that safety property. The idea is that with a proof that the monitor itself is secure wrt. a suitable security view for the monitor,[4] we can derive the security of the composed system via compositionality of the security predicate, provided that

- the security views for the monitor and the target system form a proper view separation wrt. a view for the composed system,
- the monitor and the target system are well-behaved for their view wrt. the view of the other component, and
- the side conditions for the compositionality of the desired security predicate are satisfied.

In this paper, we consider not a single target system, but we aim to find sufficient conditions under which the composition of the monitor with arbitrary target systems preserves security. We approach this problem by focusing on the monitor first and searching for sufficient conditions on the security view that guarantee that the monitor is well-behaved and secure. These conditions give rise to a *set* of views for potential target systems and corresponding views for the monitor:

Definition 4. *Let P be a safety property, ES_P be a monitor for P and SP be a security predicate. A* view-aware monitor *for P is a tuple $(ES_P, \mathcal{V}\mathbf{s}_P, \pi_P)$, where $\mathcal{V}\mathbf{s}_P$ is a set of views for potential target systems ES, and π_P is a function from views for target systems to views for the monitor. A view-aware monitor is*

- valid *if for every view $\mathcal{V} \in \mathcal{V}\mathbf{s}_P$, it holds that $\mathcal{V}$ is valid, $\pi_P(\mathcal{V})$ is valid for ES_P, and $\mathcal{V}$ and $\pi_P(\mathcal{V})$ form a proper view separation for some $\mathcal{V}'$.*
- well-behaved *if for every $\mathcal{V} \in \mathcal{V}\mathbf{s}_P$, ES_P is well-behaved for $\pi_P(\mathcal{V})$ wrt. $\mathcal{V}$.*
- secure wrt. SP *if for every $\mathcal{V} \in \mathcal{V}\mathbf{s}_P$, ES_P satisfies SP for $\pi_P(\mathcal{V})$.*

Intuitively, a view-aware monitor is enriched with a set of compatible security views for potential target systems and corresponding views for the monitor. Once we have shown a view-aware monitor for P to be well-behaved and secure wrt. a security predicate SP, and we have a concrete target system at hand that satisfies SP wrt. a compatible view $\mathcal{V} \in \mathcal{V}\mathbf{s}_P$, we just have to show the remaining conditions on the target system: that ES is well-behaved for $\mathcal{V}$ wrt. $\pi_P(\mathcal{V})$, and the side conditions for the compositionality of SP are satisfied. The resulting

[4] Which is typically different to the view for the target system because it is restricted to the set of relevant monitor events, and because monitored events that are neutral N-events for the target system have to be considered confidential C-events for the monitor (or vice versa) due to the constraints in the definition of proper view separation (particularly $N_i \cap N_j = \emptyset$, i.e. an event cannot be used for corrections in both components). See Section 3.2 for an example where this plays a role.

composed system satisfies *both* the safety property P (by Lemma 1) and the security predicate SP wrt. a view $\mathcal{V}'$, for which $\mathcal{V}$ and $\pi_P(\mathcal{V})$ form a proper view separation (by Corollary 1). As a trivial example, a monitor for an arbitrary safety property is well-behaved and secure wrt. (almost) any security predicate if the relevant events are all confidential or all visible in the target system.

Theorem 1. *Let $ES_P = (E_P, I_P, O_P, Tr_P)$ be a monitor for a safety property P. The view-aware monitor $(ES_P, \mathcal{V}\mathbf{s}_P, \pi_P)$ with*

$$\mathcal{V}\mathbf{s}_P = \{(V,N,C) \mid valid\,((V,N,C)) \wedge E_P \subseteq V \vee E_P \subseteq C\}$$

$$\pi_P\,((V,N,C)) = (V \cap E_P, \emptyset, C \cap E_P)$$

is valid, well-behaved and secure wrt. BSD and $FCIA^{\rho,\Gamma}$, and it is secure wrt. $BSIA^{\rho}$ if $\rho(\pi_P(\mathcal{V})) \supseteq C \cap E_P$ for any $\mathcal{V} \in \mathcal{V}\mathbf{s}_P$.

This follows directly from Theorems 3.5.7 and 3.5.16 of [12] about trivially satisfied BSPs and the fact that well-behavedness is trivially satisfied if there are no shared N-events. We now illustrate our approach with two more specific examples of safety properties, namely separation of duties between activities in a workflow system, and the enforcement of ordered delivery of messages by an asynchronous communication platform.

3.1 Separation of Duty

We have seen in Example 3 how to formalize separation of duty as a safety property. We could enforce this property using a monitor with event set $E_{SoD}^{a,a'} = (E_a \cup E_{a'}) \cap E_U$ and the traces in $P_{SoD}^{a,a'}$ projected onto these events. However, it is useful to refine our monitor by adding two parameters that give us more flexibility for formulating conditions for security.

- We designate a set $E^{assign} \subseteq E_{SoD}^{a,a'}$ of events that are used to assign a user to an activity. The monitor then enforces that a single user is not assigned to both a and a', and that any interaction between a user and an activity is only allowed to happen after that user has been assigned to the activity.
- The set $E^{disabled} \subseteq E_{SoD}^{a,a'}$ contains events that do not occur at runtime at all. This can be used to make explicit static knowledge of disabled events, e.g. a subset of users not being allowed to perform certain actions.

Lemma 2. *The event system $ES_{SoD}^{a,a'} = \left(E_{SoD}^{a,a'}, \emptyset, \emptyset, Tr_{SoD}^{a,a'}\right)$ is a monitor for $P_{SoD}^{a,a'}$, where $E_{SoD}^{a,a'} = (E_a \cup E_{a'}) \cap E_U$ and*

$$\begin{aligned} Tr_{SoD}^{a,a'} = \big\{ \alpha \in \left(E_{SoD}^{a,a'}\right)^* \mid\ & \forall u, u' \in U.\ \forall e, e' \in \mathrm{set}(\alpha). \\ & ((e \in (E_a \cap E_u) \wedge e' \in (E_{a'} \cap E_{u'}) \longrightarrow u \neq u') \\ & \wedge \left(\mathrm{set}(\alpha) \cap (E_a \cap E_u \cap E^{assign}) = \emptyset \longrightarrow \mathrm{set}(\alpha) \cap (E_a \cap E_u) = \emptyset\right) \\ & \wedge \left(\mathrm{set}(\alpha) \cap E^{disabled} = \emptyset\right) \big\} \end{aligned}$$

This follows from $Tr^{a,a'}_{SoD} \subseteq P^{a,a'}_{SoD}$ and $E^{a,a'}_{SoD}$ being a relevant event set. We can show that this monitor satisfies BSD, $BSIA$ and $FCIA$ if

- user assignment is non-confidential, or
- only confidential or only visible user interaction events are enabled, or
- the separation of duty constraint is enforced statically (i.e. the sets of users for whom interaction events with a and a' are enabled, respectively, are disjoint) and dynamic user assignment is permissive (i.e. $E^{assign} = E^{a,a'}_{SoD}$).

Formally, these conditions are captured in $\mathcal{V}\mathbf{s}^{a,a'}_{SoD}$ of the following definition:

Lemma 3. *The view-aware monitor* $\left(ES^{a,a'}_{SoD}, \mathcal{V}\mathbf{s}^{a,a'}_{SoD}, \pi^{a,a'}_{SoD}\right)$ *with*

$$\begin{aligned}\mathcal{V}\mathbf{s}^{a,a'}_{SoD} = \big\{(V,N,C) \mid\ & valid\,((V,N,C)) \wedge E^{a,a'}_{SoD} \subseteq V \cup C \\ & \wedge \big(E^{assign} \subseteq V \\ & \quad \vee \left(V \cap E^{a,a'}_{SoD} \subseteq E^{disabled} \vee C \cap E^{a,a'}_{SoD} \subseteq E^{disabled}\right) \\ & \quad \vee \left(\mathrm{users}(E_a) \cap \mathrm{users}(E_{a'}) = \emptyset \wedge E^{assign} = E^{a,a'}_{SoD}\right)\big\}\end{aligned}$$

$$\pi^{a,a'}_{SoD}\left((V,N,C)\right) = \left(V \cap E^{a,a'}_{SoD}, \emptyset, C \cap E^{a,a'}_{SoD}\right)$$

where $\mathrm{users}(E) = \left\{u \in U \mid \exists e \in \left((E \setminus E^{disabled}) \cap E_u\right)\right\}$*, is valid, well-behaved and secure wrt. BSD, BSIA*$^{\rho}$ *and FCIA*$^{\rho,\Gamma}$ *if* $\rho(\mathcal{V}) \supseteq E^{assign}$.

Due to space constraints, we place the proofs of this and the following lemmas and theorems into an extended version of this paper [2].

For this monitor, the security predicates BSD, $BSIA^{\rho}$ and $FCIA^{\rho,\Gamma}$ (for suitable ρ) are preserved upon composition as follows:

Theorem 2. *Let* $ES = (E, I, O, Tr)$ *be an event system and* $\mathcal{V} \in \mathcal{V}\mathbf{s}^{a,a'}_{SoD}$ *be a view for ES. Then*

- $BSD_{\mathcal{V}}(Tr)$ *implies* $BSD_{\mathcal{V}}(Tr(ES \| ES^{a,a'}_{SoD}))$*, and*
- $BSD_{\mathcal{V}}(Tr) \wedge BSIA^{\rho}_{\mathcal{V}}(Tr)$ *implies* $BSIA^{\rho}_{\mathcal{V}}(Tr(ES \| ES^{a,a'}_{SoD}))$ *if* $\rho(\mathcal{V}) \supseteq E^{assign}$.
- $BSD_{\mathcal{V}}(Tr) \wedge BSIA^{\rho}_{\mathcal{V}}(Tr) \wedge FCIA^{\rho,\Gamma}_{\mathcal{V}}(Tr)$ *implies* $FCIA^{\rho,\Gamma}_{\mathcal{V}}(Tr(ES \| ES^{a,a'}_{SoD}))$ *if* $\rho(\mathcal{V}) \supseteq E^{assign}$.

This means that if the target system satisfies one of the above combinations of security predicates, then the monitored system $ES \| ES^{a,a'}_{SoD}$ still satisfies it, and it additionally satisfies the separation of duty property (by Lemma 1).

Example 4. In our workflow scenario, we only considered the values of data items confidential, not the identity of participants in the workflow. We therefore simply used the events of the form $\mathrm{Start}_a(u)$ as assignment events and chose a view that considers these events as visible. Hence, the case $E^{assign} \subseteq V$ applies[5] and we can use Theorem 2 for the security-preserving enforcement of arbitrary separation of duty constraints. □

[5] Note that $E^{assign} \subseteq V$ does not mean that these events *have to be* visible for an observer of the system, it just means that if we are able to prove security wrt. this

3.2 Ordered Delivery of Asynchronous Messages

Another safety property we encountered while working on [3] is the guarantee of ordered delivery of messages by the asynchronous communication platform. When we specified our workflow system in terms of communicating subsystems in [3], we did not include any guarantees regarding message delivery in the specification of the communication platform. This simplified the specification of the platform and the proof of compositionality, but it made the specifications of the communicating subsystems more complex. We had to introduce explicit acknowledgment messages and make the subsystems wait for acknowledgments before continuing with a communication protocol in some cases. Message delivery ordering per sender-receiver pair, i.e. the guarantee that messages between two components are received in the order that they are sent, makes these explicit acknowledgments unnecessary in the cases we encountered. It turns out that we can use the same compositional approach as above to analyze the impact that this refinement of the communication platform has on the requirements regarding information flow.

We first formulate ordered delivery as a safety property. Let $\mathrm{sentMsgs}(a, b, \alpha)$ and $\mathrm{rcvdMsgs}(b, a, \alpha)$ denote the sequences of messages m contained in the sequences of events of the form $\mathrm{Send_a(b, m)}$ or $\mathrm{Recv}_b(a, m)$, respectively, in a trace α, and let $\preceq$ be the prefix order on traces. Ordered delivery can be formulated as

$$Tr_{CD} = \{\alpha \mid \forall a, b.\ \mathrm{rcvdMsgs}(b, a, \alpha) \preceq \mathrm{sentMsgs}(a, b, \alpha)\}$$

The event system $ES_{CD} = (E_{CD}, \emptyset, \emptyset, Tr_{CD})$ with the relevant set of events

$$E_{CD} = \{e \mid \exists a, b, m.\ e = \mathrm{Send}_a(b, m) \vee e = \mathrm{Recv}_b(a, m)\}$$

is a monitor for ordered delivery, assuming communication between components is represented by Send and Recv events of the form given above.

It turns out that, in order for the refined communication platform to be secure, we have to treat Recv events corresponding to confidential Send events as N-events. The reason is that we might have to correct the deletion or insertion of a confidential Send event by removing or inserting a corresponding Recv event at the correct position in the trace in order to preserve the correct order of delivery.

Lemma 4. *The view-aware monitor* $(ES_{CD}, \mathcal{V}\mathbf{s}_{CD}, \pi_{CD})$ *with*

$$\begin{aligned}\mathcal{V}\mathbf{s}_{CD} = \big\{(V, N, C) \mid\ & valid\,((V, N, C)) \wedge E_{CD} \subseteq V \cup N \cup C \\ & \wedge Send_a(b, m) \in V \longleftrightarrow Recv_b(a, m) \in V \\ & \wedge Send_a(b, m) \notin V \longleftrightarrow Recv_b(a, m) \in C\big\}\end{aligned}$$

$$N_{CD}(C) = C \cap \{e \mid \exists a, b, m.\ e = Recv_a(b, m)\}$$

view, then the system is secure *even if* user assignment were visible for an observer. This notion of strengthening views is captured formally in Theorem 1 of [11], for example.

$$\pi_{CD}\left((V,N,C)\right) = (V \cap E_{CD}, N_{CD}(C), E_{CD} \setminus (V \cup N_{CD}(C)))$$

is valid, well-behaved and secure for BSD *and* $BSIA^{\rho}$ *and* $FCIA^{\rho,\Gamma_{CD}}$ *for any* ρ *and* $\Gamma_{CD} = (\nabla_{CD}, \Delta_{CD}, \Upsilon_{CD}) = (E_{CD}, \emptyset, E_{CD})$.

In this case, there are further side conditions on the target system that follow directly from the requirement of well-behavedness. Moreover, confidential Recv events become neutral in the security view of the composed system. Confidential Send events, however, and thus the message contents, remain confidential.

Theorem 3. *Let* $ES = (E, I, O, Tr)$ *be an event system and* $\mathcal{V} = (V, N, C)$ *be a view for* ES *such that*

- $\mathcal{V} \in \mathcal{V}\mathbf{s}_{CD}$, *and*
- $total(ES, N_{CD}(C))$, *and*
- $BSD_{\mathcal{V}}(Tr) \wedge BSIA^{\rho}_{\mathcal{V}}(Tr)$ *holds for some* ρ, *and*
- $N \cap E_{CD} \neq \emptyset$ *implies* $FCIA^{\rho,\Gamma}_{\mathcal{V}}$ *for some* $\Gamma = (\nabla, \Delta, \Upsilon)$ *with* $E_{CD} \subseteq \nabla$, $E_{CD} \subseteq \Upsilon$, *and* $E_{CD} \cap \Delta = \emptyset$.

Then $BSD_{\mathcal{V}'}(Tr(ES \| ES_{CD})) \wedge BSIA^{\rho}_{\mathcal{V}'}(Tr(ES \| ES_{CD}))$ *holds for* $\mathcal{V}' = (V, N \cup N_{CD}(C), C \setminus N_{CD}(C))$, *and* $FCIA^{\rho,\Gamma}_{\mathcal{V}}(Tr)$ *implies* $FCIA^{\rho,\Gamma}_{\mathcal{V}'}(Tr(ES \| ES_{CD}))$.

Example 5. In [3], we have already proven BSD and $BSIA$ for our workflow system wrt. a view such that most of the preconditions of Theorem 3 are satisfied, i.e. non-visible Recv events are treated as confidential and are accepted at any time by the individual subsystems. However, we had to use some Send events for corrections in our proofs. Hence, $N \cap E_{CD} \neq \emptyset$ holds and in order to apply Theorem 3, we get $FCIA$ as an additional proof obligation.[6] As $FCIA$ is relatively similar to $BSIA$ and we had already proven $BSIA$ for the activities in our example workflow, it turns out to be easy to prove in this case. □

4 Related Work

The connection between safety properties and execution monitors is elaborated in [17]. Information flow security is of a different nature than safety properties. In [8], possibilistic information flow properties are characterized as closure properties on the whole sets of traces of a system. Hence, removing traces in order to enforce a safety property can invalidate such a closure property. This explains the refinement paradox, which was already observed in early works such as [7].

The idea of using composition for the security-preserving enforcement of safety properties also occurs in [14, Section 3.2] for the framework of McLean's selective interleaving functions. We apply and elaborate this idea in the context of the MAKS framework [8], which has been shown to be more expressive than

[6] Intuitively, this means that we may use Send events for corrections, but not in direct response to the insertion of a Recv event, in order to avoid a non-terminating sequence of communication events. See [12, pages 132f] for a discussion of this issue.

McLean's framework [13]. We demonstrate the approach by deriving three results giving explicit and succinct conditions for the security-preserving enforcement of safety properties (Theorems 1 to 3). For this purpose, we heavily rely on the well-developed MAKS framework, in particular its compositionality results [11].

In the context of MAKS, a paper with a goal very similar to ours is [10]. The approach is different, however. It requires a proof of security of the target system via unwinding, and then modifies the safety property to be enforced by removing or adding traces so that the unwinding conditions are preserved. It works with arbitrary safety properties, but the result can be hard to predict, as it depends heavily on the unwinding relation that is used. We see this approach as complementary to ours. It can be used if compatibility results as we presented them above are not available for the safety property in question.

There are approaches for security-preserving process refinement (i.e. reducing the set of possible traces) also for other notions of information flow security. [16] considers confidentiality-preserving refinement for probabilistic information flow. [18] builds upon the MAKS framework, but modifies the notions of system specification and security predicates to make the distinction between underspecification and unpredictability explicit. [4] uses a similar approach to [10], but in the context of a process algebra and bisimulation-based notions of security. Which of the available approaches is best suited for a concrete application depends on the precise security requirements at hand.

5 Conclusion

In this paper, we have focused on the compatibility of possibilistic information flow security and safety properties. We have described how existing compositionality results for information flow predicates can be used to derive sufficient conditions for compatibility with a given safety property. We found this approach to be useful in our case study of verifying the specification of a distributed workflow management system [3].

While Theorem 1 applies to arbitrary safety properties, results like our Theorems 2 and 3 have to be derived for each safety property of interest individually. However, it is worth pointing out that the compatibility result for separation of duty is parametric in the event sets and can therefore be instantiated for arbitrary systems where users participate in distinct activities in the presence of separation of duty constraints. Similarly, ordered delivery can be applied to any system with asynchronous message passing. This demonstrates that compositional reasoning can be used to derive compatibility results for whole classes of common safety properties.

In this paper, we have considered systems and properties on a high level of abstraction. In order to move to a more concrete level of implementation detail, we intend to focus on action refinement in future work. Combined with the compositional reasoning described in this paper and in [3], this facilitates a step-wise development process. Eventually, we hope to integrate these techniques into a development tool for provably secure workflow management systems.

Acknowledgments We thank Richard Gay, Sylvia Grewe, Steffen Lortz, Heiko Mantel and Henning Sudbrock for providing a formalization of the MAKS framework in Isabelle/HOL that allowed us to verify our main results in Isabelle, and the anonymous reviewers for helpful comments.

References

1. Alpern, B., Schneider, F.B.: Recognizing safety and liveness. Distributed Computing 2(3), 117–126 (1987)
2. Bauereiss, T., Hutter, D.: Compatibility of safety properties and possibilistic information flow security in MAKS. Tech. rep. (2014), available at `http://bauereiss.name/papers/SEC2014_TR.pdf`
3. Bauereiss, T., Hutter, D.: Possibilistic information flow security of workflow management systems. In: GraMSec'14, to appear in EPTCS (2014)
4. Bossi, A., Focardi, R., Piazza, C., Rossi, S.: Refinement operators and information flow security. In: SEFM. pp. 44–53. IEEE Computer Society (2003)
5. Clarkson, M.R., Schneider, F.B.: Hyperproperties. Journal of Computer Security 18(6), 1157–1210 (2010)
6. Hutter, D., Mantel, H., Schaefer, I., Schairer, A.: Security of multi-agent systems: A case study on comparison shopping. J. Applied Logic 5(2), 303–332 (2007)
7. Jacob, J.: On the derivation of secure components. In: IEEE Symposium on Security and Privacy. pp. 242–247. IEEE Computer Society (1989)
8. Mantel, H.: Possibilistic definitions of security - an assembly kit. In: CSFW. pp. 185–199. IEEE Computer Society (2000)
9. Mantel, H.: Unwinding possibilistic security properties. In: Cuppens, F., Deswarte, Y., Gollmann, D., Waidner, M. (eds.) ESORICS. LNCS, vol. 1895, pp. 238–254. Springer (2000)
10. Mantel, H.: Preserving information flow properties under refinement. In: IEEE Symposium on Security and Privacy. pp. 78–91. IEEE Computer Society (2001)
11. Mantel, H.: On the composition of secure systems. In: IEEE Symposium on Security and Privacy. pp. 88–101. IEEE Computer Society (2002)
12. Mantel, H.: A uniform framework for the formal specification and verification of information flow security. Ph.D. thesis (2004)
13. Mantel, H.: The framework of selective interleaving functions and the modular assembly kit. In: Atluri, V., Samarati, P., Küsters, R., Mitchell, J.C. (eds.) FMSE. pp. 53–62. ACM (2005)
14. McLean, J.: A general theory of composition for a class of "possibilistic" properties. IEEE Trans. Software Eng. 22(1), 53–67 (1996)
15. Roscoe, A.: Parallel operators. In: Understanding Concurrent Systems, pp. 45–66. Texts in Computer Science, Springer London (2010)
16. Santen, T.: Preservation of probabilistic information flow under refinement. Inf. Comput. 206(2-4), 213–249 (2008)
17. Schneider, F.B.: Enforceable security policies. ACM Trans. Inf. Syst. Secur. 3(1), 30–50 (2000)
18. Seehusen, F., Stølen, K.: Maintaining information flow security under refinement and transformation. In: Dimitrakos, T., Martinelli, F., Ryan, P.Y.A., Schneider, S.A. (eds.) Formal Aspects in Security and Trust. LNCS, vol. 4691, pp. 143–157. Springer (2006)